

Mission Planning System for the TET-1 OnOrbitVerification Mission

Andreas Spörl* Christoph Lenzen* Maria Theresia Wörle* Jens Hartung*
Falk Mrowka† Armin Braun‡ Martin Wickler§

*Deutsches Zentrum für Luft- und Raumfahrt e. V., German Aerospace Center
Münchener Straße 20, 82234 Weßling, Germany*

The TET-1 satellite was launched on July 22nd, 2012, to test and demonstrate the space readiness of new hardware components. Eleven experiments are running in space since then. The mission planning system (MPS) that provides the TET-1 satellite^{1,2} with its tele-command timelines during the OnOrbitVerification (OOV) phase is presented: Based on a strategic one-year experiment plan provided in advance by an external industry partner, MPS collects all relevant information necessary to build a sequence of flight procedures, called timeline, for a time range of roughly a week, on a day-by-day basis. In contrast to the TerraSAR-X/TanDEM-X MPS³ or the Incremental Planning System,⁴ where several software components convert incoming orders into commandable files, a slim set of tools was decided to be used for the TET-1 mission, combined in PINTA (Program for INteractive Timeline Analysis). Necessary data was imported using the plug-in mechanism of PINTA that uses interfaces to several partners. Having all information available, scheduling itself was done by running the planning algorithms provided by Plato, GSOC's generic library for modeling and solving planning problems. An assembly of various planning algorithms, individually configurable and referencing one another,⁷ creates the necessary timeline entries of flight procedures. Due to the high flexibility of the planning system it was possible to support various changes in the pre-planned onboard timeline on short notice.

Additionally, an outlook on further extensions of the current MPS is given, that enables even more flexibility in terms of data acquisition and are relevant for the upcoming FireBIRD mission, which includes the TET-1 spacecraft after the OOV operations phase.

I. Introduction

The main input for the MPS of TET-1 in the OOV mission^{1,2} was a sheet with several weekly scenarios, containing all experiments and their pre-planned execution times. The external partner who provided this sheet was responsible for all experiments and considered all their constraints for the preparation of this pre-plan, like memory, power, lighting conditions etc.

However, due to certain additional constraints reasoned by the satellite bus, e.g. the necessity of cooling days, it was not possible to stick to the execution of the weekly scenarios as-is. They were splitted into day-by-day scenarios.

Next to experiment days additional cooling days had to be introduced to give the satellite time to get rid of superfluous thermal energy. Initially seen as a drawback, the cooling days turned out to be helpful at a stage of the mission when certain experiments needed more freedom in selecting their execution time.

Figure 1 on the following page summarizes roughly one week of operations on TET-1.

*Mission Planning System Engineer, Mission Operations Department, German Space Operations Center

†Head of Mission Planning Team, Mission Operations Department, German Space Operations Center

‡Head of Mission Control and Data Systems Team, Mission Operations Department, German Space Operations Center

§Deputy Head of Mission Operations Department, Mission Operations Department, German Space Operations Center

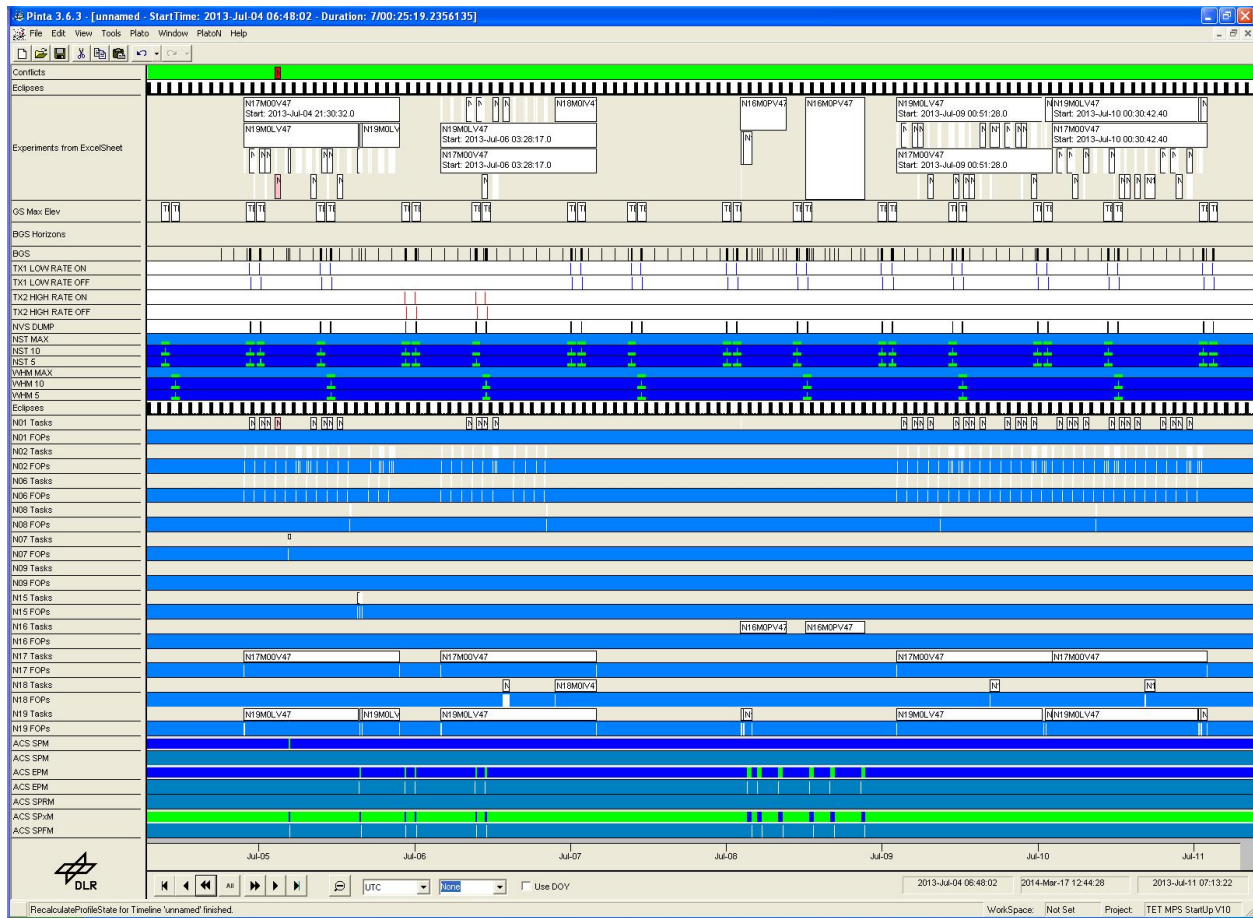


Figure 1. Roughly one week of payload operations. The plot shows five days of payload operations, including a cooling day. The first line indicates conflicts in the timeline under investigation. The second line shows lighting conditions, i.e. sunlight and shadow phases. The third line shows all experiments in the current timeline. Daily scenarios are represented by blocks of length one day. Nominally the daily scenarios adjoin to one another like the last two in the plot. The introduction of cooling days allowed to move scenarios by full orbits, as it can be seen for the second daily scenario which was actually followed by a period without experiment execution. Below that, plotted on white, high- and low-rate transmitter switches of the background sequence are shown. An overview of scheduled ground stations is plotted in green and blue. Task plots for all individual experiments and their scheduled flight procedures make up the main part of the figure. Finally, information about attitude mode is plotted, indicating high-rate ground station contacts or certain experiment executions, for example N15 infrared camera datatakes.

II. TET OOV Mission Planning Task

The general planning task was to provide the Flight Operations Team with a conflict-free timeline that contains all flight procedures, i.e. a sequence of time tagged tele-commands, to run the OOV mission on TET-1 for one week. There were eleven experiments on board, namely

- N1: LithiumPolymer Battery
- N2: Flexible Thin layer solar array
- N6: Sensorbus System and Power Drain
- N7: Pico Satellite Propulsion Experiment
- N8: Next Generation Solar Array
- N9: Next Generation Solar Array

- N15: Infrared camera system
- N16: TwoFrequencies GPS
- N17: HardwareBOSS
- N18: Keramis Transponder experiment
- N19: Memory Orbit Radiation Experiment

Some experiments were not commanded within in the MPS, one of it being N9: a solar array that continuously ran and very rarely required special commands. These commands were prepared manually by system engineers, the same holds for N18, the fly-by experiments. MPS provided envelopes for the latter, so no other experiment execution took place in parallel. Furthermore access to the satellite (e.g. for tele-command uplink or data downlink) was blocked when these experiments were running.

The experiments during one planning week were grouped to daily scenarios. A daily scenario contains a sequence that is assured to be conflict-free with respect to constraints like memory, downlink capability, power consumption and thermal constraints. Experiments were ordered by their execution time and, if available, their mode, which indicates the configuration of the experiment.

The sequence of scenarios had been provided in advance by the external industry partner on a weekly basis. Given this sequence, MPS collected all other relevant data like orbit events and ground station scheduling to complete the necessary data for the planning run. Some experiments needed further special treatment, to ensure conflict-free experiment execution or meaningful data acquisition. These manipulations have been done in two ways: Where simple rules applied, a special repair algorithm has been used, e.g. for shifting the battery charging experiments according to their constraint on lighting conditions. On the other hand, manual modifications have been performed with respect to IR camera experiments, where image acquisition experiments were shifted in time in order to acquire data over points of interest.

This experiment timeline was complemented with flight procedures supporting ground station contacts, i.e. the background sequences, which cover attitude switches, transmitter switches, requests for extended housekeeping data, and requests for data dumps. The background sequence is so-to-say the backbone of the mission timeline, containing repetitive tasks that have to be performed to ensure a continuous mission. Since there were several types of background sequences, subsets of these tasks contribute to fit the needs of a given type.

The overall workflow is shown in Figure 2. It contains the following three main steps:

1. Collecting necessary data from preplanning, flight dynamics and ground station scheduling office
2. Scheduling of all tasks for a given timeline horizon, taking into account all modeled constraints
3. Exporting of the resulting execution timeline into a file, containing all time-tagged tele-commands

III. GSOC MPS Generic Tools

In the following section, we given an overview over the functionalities of PINTA, GSOC's interactive, visual planning project and timeline editor, including its generic modeling language⁵ and its automated scheduling capabilities, provided by PLATO, GSOC's scheduling engine.⁸

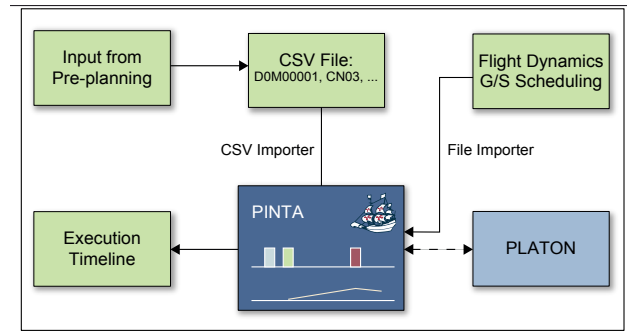


Figure 2. The workflow of the MPS is shown. The pre-scheduled timeline for the whole year of experiment execution was exported from the Excel sheet into week-based timeline scenarios. These were further split into daily scenarios. A project-specific plug-in allowed to import these scenarios into the planning system. In addition, information about orbit events and ground station scheduling is imported. The whole data model is provided to the planning engine Plato that returns a conflict-free timeline. This is finally exported into an xml file containing all time-tagged tele-commands.

A. General Data model

The data model⁵ supports a hierarchical structure, an example is shown in Figure 3. On top level, i.e. directly below Task Tree, you may define groups and tasks, however the latter is uncommon as all tasks should be sorted within some category, usually represented by a group.

A group may contain any number of child-groups and child-tasks. It may be given bounds on the number of elements which must be scheduled and it is given the information when itself is considered to be scheduled.

Tasks on the other hand can be given timeline entries, i.e. time intervals where they are planned to execute. They represent the granularity of the planning project in the sense that the sequence of commands (or other sub-activities), which are represented by the task, must be completely specified by the start- and end-time of a task's timeline entry. A task contains information about its desired duration, which may be considered by the scheduling algorithm. Furthermore a task may be given constraints such as minimal and maximal duration, time dependencies to other tasks and resource dependencies: A Comparing Resource Dependency defines local bounds on the resource, i.e. it specifies that a resource must have a certain value during the execution of the task. When the resource is consumed or provided by the task execution, one can define an Allocating Resource Dependency, where the resource modification is restricted to the time interval where the task is executed, or one can define an Accumulating Resource Dependency, in which case the consumption persists. Whenever a task changes a resource value to above or below the global bounds that are allowed for the given resource and whenever the resource's profile exceeds a local bound generated by a timeline entry and a Comparing Resource Dependency to its task, conflicts are indicated. All resource dependencies may be given offsets and profiles rather than constant values. When generating a timeline entry, the profile is mapped to the time axis according to the time interval defined by the timeline entry and the offsets of the constraint.

Among others not in use, these are the main features of the GSOC modeling language⁵ that were utilized by the TET MPS.

B. Overview of Basic Algorithms

Each planning run is initially restricted to the planning horizon of interest, which, in our case, is the week that has to be scheduled. The overall algorithm is a sequence of several linked and interleaved subalgorithms. Every subalgorithm might further contain a sequence of subalgorithms and so on, until it reaches a level of basic algorithms. The main basic algorithms are shortly explained in subsequent sections.

1. StaticSchedulingRules

StaticSchedulingRules can be used to schedule any task according to the existence of an opportunity where a resource value attains a configurable value. Once such an opportunity is found in the timeline, a sequence of tasks may be scheduled with configurable offsets. An opportunity may also be a merged opportunity with maximum gaps between resource availabilities. These gaps may also be configured.

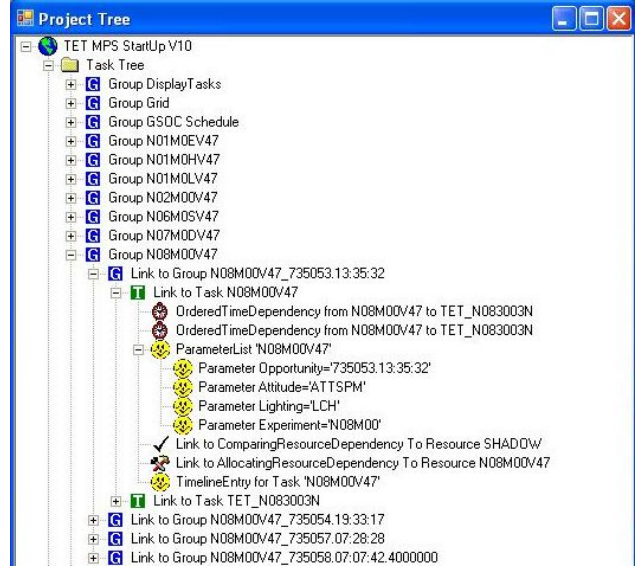


Figure 3. The hierarchy in the data model. Several groups of experiments, abbreviated like N08M00V47 are displayed. The notation N08 is the experiment, M00 is further information about the mode of the experiment, V47 is the version of the prescheduled excel sheet provided by the external partner. A given experiment group contains all execution tasks of this experiment as subgroups. Within every subgroup there is one task representing the experiment and one or more tasks representing the necessary flight procedures, called TET..., to execute the task. For each task, links to resource comparisons or modifications, ordered time dependencies, parameter lists and timeline entries are shown.

This algorithm type was mainly used for scheduling background sequence tasks. For example, transmitter switches are scheduled if the corresponding scheduled ground station resources are available.

2. Filter Algorithm

In addition to scheduling tasks via detection of resource availabilities, it is also possible to select objects in a configurable order and to schedule a selected task according to configurable rules, the most important of course that all constraints shall be obeyed. Of course there also exist other filters than this NoConflict filter, but for the TET-OOV mission, it was sufficient to select the first possible timeline entry, provided that the experiments were selected in the proper order. For this purpose the FirstInTime object filter has been designed, this way scheduling starts with the object, which has no unscheduled predecessor. For example see the OrderedTimeDependency in Figure 3 on the previous page between the experiment task N08M00V47 and the flight procedure TET_N083003N.

IV. Modeling of TET OOV Mission planning problem

The model containing all all tasks and constraints may be split into two main parts: An experiments part and a part that is related to the background sequence generation. This splitting was chosen since background sequence events like ground station contacts could be handled all at once using recurring tasks, whereas every experiment should be handled separately by the planning algorithm. So the latter had one experiment task for each single experiment execution. Also the algorithms that scheduled the flight procedures are different.

All data were imported into a datamodel, that only contains some basic configurations, like resource bounds or recurring tasks. There were essentially two kinds of information that were imported, as shown in the following sections.

A. Orbit Events and Ground Station Contacts

The flight dynamics department provide information about elevations over ground stations, satellite eclipses and equator crossings. In order to perform the scheduling of background sequence tasks it is necessary to gather information about access to ground station networks and ground station visibilities. Once all these information are imported into the mission planning model, the timeline entries of background sequence tasks can be derived.

1. Scheduled Ground Station Passes and Ground Station Elevation

For a satellite ground station contact two conditions must be met: the ground station has to be prepared for the contact and the satellite must be seen above the horizon.

The information about which passes are agreed upon is sent to MPS via an interface item from the ground station scheduling office. Since the scheduling of passes may have happened weeks before the actual contact and the orbit of the satellite may have changed meanwhile, the information about scheduled passes is used as an envelope with extended start and stop times.

The information about the elevation of the satellite above the ground station's horizon during the contact as well as the exact timing information is provided by the Flight Dynamics (FD) department.

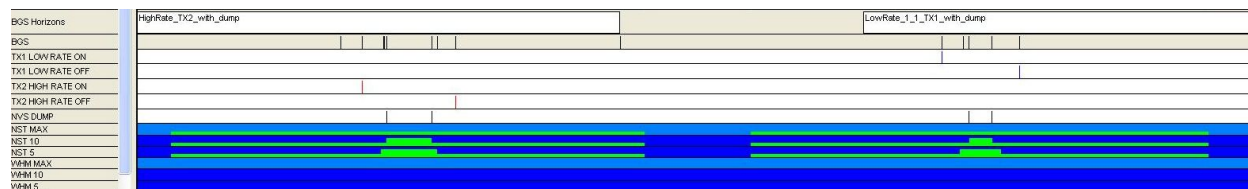


Figure 4. Background sequences and their horizons. The plots on top show background sequence horizon tasks which are used to trigger the scheduling of a certain type of background sequence. The plots below show scheduled flight procedures to actually perform the ground station contact activities. The resource plots in blue and green show the envelopes of the scheduled ground stations and the 5° respectively 10° elevation events.

Both, flight dynamics and ground station scheduling information, contribute to a common set of resources to identify scheduled ground station contacts with the latest information about start and stop times. These combined resources for ground station contacts were set up for 5° and 10° degrees of elevation over the ground stations horizon. Where both contributions are present, the ground station contact resources indicate an opportunity only for time frames during which one station is scheduled along with one orbit fly-by of the satellite.

Figure 4 on the preceding page displays the situation with two examples.

2. *Shadow and Equator crossings*

Another information provided by FD are the times when the satellite is in eclipse, including eclipses by the moon, too. This information is necessary to check for conflicts of experiments relying on certain lighting conditions. Furthermore the information about equator crossing is imported and used at a later stage during the mission planning workflow to calculate the actual orbit duration.

B. Experiments

The structure of the daily scenarios, that are imported by the MPS, i.e. PINTA, contained the following information about each experiment:

- experiment identifier and mode of the experiment, if applicable
- day of year (was not used anymore, after switching to daily schedule)
- orbit of execution
- start time of experiment relative to orbit start
- comments

As indicated, for several experiments there was a set of different modes that additionally specified how the experiment should be performed respectively. For example the transponder experiment N18 had four different experiment modes N18I, N18T, N18D and N18U.

Every experiment day started with the first sunrise for the satellite after midnight. The start time for each individual experiment was calculated by multiplying the orbit of execution by the orbit duration and adding the start time relative to the orbit start time. Each of these experiment days comprised 15 orbits each lasting 95 minutes, resulting in 23 hours and 45 minutes. The gap of 15 minutes, posed some problems on long-term experiments, which will be dealt with later. Further the assumed 95 minutes had to be replaced by the real orbit duration. This was calculated by subtracting subsequent equator crossing events.

For each experiment that should be executed at a certain time, given by the pre-planning input, there exists a group that contains the following tasks: the experiment task itself and necessary flight procedures, to actually execute the experiment onboard the satellite. The groups and tasks for a given experiment are generated by the import plug-in. Resource dependencies are created in two steps: 1. OrderedTimeDependencies are created while importing the daily experiment scenarios, 2. other resource dependencies are created by the ConstraintGenerator, another plug-in algorithm to create all kinds of necessary constraints, configured via xml files.

Each experiment task has an AllocatingResourceDependency on a experiment indicator resource in order to visually display that the experiment actually runs and for conflict checking, as follows. To ensure that an experiment will meet its conditions ComparingResourceDependencies between the experiment and the according resources are used. It is possible to indicate conflicts, for example caused by forbidden experiment overlaps or wrong lighting conditions. The specific conditions to be ensured had been provided to MPS during the design phase, but due to the high flexibility achieved with software configuration it was possible to extend or reduce constraints throughout the whole mission.

The time relations between the flight procedure and the experiments are represented by OrderedTimeDependencies. They guarantee that the experiments are commanded correctly. OrderedTimeDependencies are chosen such to uniquely identify a specific point in time for a flight procedure to be executed, once the experiment itself is put on the timeline during import of the daily scenario.

For most of the experiments, it was possible to combine all necessary commands to one single flight procedure that started the experiment at a specified time and stopped it as well, or a small set of flight procedures containing commands to start, stop and configure an experiment.

Two slightly more complicated experiments which were of further interest during the planning run, and thus the scheduling algorithm setup, are listed below in order to demonstrate how we dealt with experiments that needed more than just a single flight procedure.

1. *N1 battery experiment*

For the battery experiment N1 it was necessary to ensure that the battery was not charged during eclipses. A corresponding ComparingResourceDependency on the shadow event indicated conflicts every time that this happened, as can be seen in Figure 5 on page 9. Furthermore it was necessary to check whether the solar panel experiments N2 and N8 were not active at the time of N1 experiment execution.

To execute the experiment itself only a single flight procedure was needed, but to avoid errors of the satellite's magnetic coil system it was also necessary to disable the usage of the magnetic coil system while the experiment was running. This additional requirement was introduced while the mission was already in orbit. It could be handled by simply adapting the configuration of the MPS via adding the according tasks and constraints to the model.

2. *N15 IR camera experiment*

Since the IR camera system also collects data in the visual part of the spectrum, it was necessary to ensure that datatakes were not executed during eclipses. Furthermore, a parallel execution of the experiments N2 and N6 had to be excluded. ComparingResourceDependencies would have indicated conflicts in these cases.

Along with the execution of N15, high-precision GPS data was to be collected. All flight procedures had to be scheduled relative to the execution of the datatake with the help of OrderedTimeDependencies.

V. The Algorithms

A. Background Sequence

The flight procedures used for the background sequence execution were picked from a pool of flight procedures. This pool was represented by a group in the data model containing all tasks of flight procedures. All background sequence algorithms addressed this group to get their necessary flight procedures.

Due to the different amounts of memory accumulated during the various experiments on board, three types of background sequences were set up: high-rate with dump, low-rate with dump and low-rate without dump. During low-rate passes without dump, no payload data were sent to the ground station. During passes with dump the transmission ratio of house keeping to payload data was 1 : 1 for low- and 1 : 15 for high-rate.

The type of background sequence used most often was the one configuring downlink with dump in low-rate. In most cases this was sufficient to submit all experiment data to ground. Using the 4π radiation pattern of the low-rate antenna (TX1), it was not necessary to change the attitude mode of the satellite for this case. Whenever more data was generated on board to be downlinked, e.g. after execution of the N15 or N16 experiment, the high-rate antenna (TX2) had to be used for a faster submission of the data. Since this antenna's pattern is more narrow than the one of the low-rate antenna, an earth-pointing attitude mode (EPM) was necessary. After the pass the spacecraft then had to be commanded back to the nominal mode, the sun-pointing fixed mode (SPFM). Finally, for proficiency passes over ground stations that were not used on a regular basis but had to be kept prepared for a support during contingencies, the third type of background sequence was used, low-rate with no dump scheduled.

Table 1 on the following page summarizes all types of background sequence activities.

It is obvious that the algorithms for all types of background sequences could be composed as a sequence of algorithms. This makes it comfortable to compose several types of background sequence algorithms out of reusable building blocks. To place the flight procedures correctly in the timeline, StaticSchedulingRules were used, configured such that the flight procedures were scheduled when the respective resource attained the corresponding value. Offsets were adjusted accordingly. Along with to the ground station contact driven activities on board, a repeating task to gather extended housekeeping data by the attitude control system of the bus was scheduled every two hours.

Table 1. The different types of background sequences and their tasks.

Start Time	High Rate with Dump	Low-Rate with Dump	Low-Rate no Dump
−06:30 rel. to start of 5°	EPM		
−03:00 rel. to start of 5°	TX2 on	TX1 on	TX1 on
+00:30 rel. to start of 5°	ext. HK	ext. HK	ext. HK
+00:00 rel. to start of 10°	Dump on	Dump on	
+00:00 rel. to end of 10°	Dump off	Dump off	
+00:01 rel. to end of 5°	SPFM		
+03:00 rel. to end of 5°	TX2 off	TX1 off	TX1 off

B. Scheduling Experiment Tasks

Other than background sequence activities for which scheduling was triggered by opportunities represented by resources, flight procedures for experiments were scheduled differently. Since each experiment needed to be handled individually, the flight procedures for each experiment were located within the group of the experiment itself. To schedule these flight procedures it was necessary to specify the correct start times. This was done with `OrderedTimeDependencies`. Then, an algorithm could loop through all the experiment groups, picked all flight procedures and located them on the timeline at the only place where it was possible without introducing conflicts. Once a group had been scheduled it was flagged such that it was not scheduled again during this planning run.

C. Special Cases

Not all experiments could be coped with with the simple approach sketched above but there were special challenges. Some of the problems that the MPS faced were caused by the assumption made for the pre-planning by the customer, that the orbit duration would stay fixed to the value of 95 minutes during the whole mission time of a whole year. Since on TET-1 no orbit control mechanisms are implemented, this wrong assumption had to be compensated by the MPS with adjustments of the plan to fit the actual times of orbital events. The orbit duration was calculated every time a new flight dynamics input was delivered by measuring the difference between the given equator crossings. This led to a mismatch between the actually needed schedule for the experiment executions and the pre-planning. Special algorithms were developed to deal with the resulting problems:

1. Due to the changing orbit it happened for example that an experiment that should take place during sunlight got moved partly into an eclipse. So this experiment needed a shifted start time.
2. Furthermore, there were two long-term experiments which were not supposed to be switched off regularly, but only at specific points in time.
3. And last, a special treatment was necessary to provide more flexibility for the execution times and thus the regions of interest for the IR camera datatakes to be executed.

These cases are detailed in the following sections.

1. N1 Repair Algorithm

An experiment execution of N1 that was erroneously placed in an eclipse had to be moved in the timeline until its execution took place in the sunlight. See Figure 5 on the next page for an example. A repair algorithm was invented to achieve this. The algorithm uses a configurable planning horizon, smaller than the overall planning horizon. Within this horizon it can be chosen to move the experiment task either to later or earlier times till it is conflict-free. After having found the new correct start time for the experiment, the according flight procedure could be scheduled in the nominal way.

2. Algorithms and Constraints with merged opportunities for N17 and N19

The two long-term experiments imposed additional requirements on the MPS since starting and stopping them was only desired at times where a longer period with no other experiment execution was foreseen in the



Figure 5. Conflicts after importing N1 operations. The conflicts appeared since the prescheduled plan assumed a fix orbit duration of 95 minutes, whereas the actual orbit duration was shorter and decreasing. The start times of the N1 experiment execution therefore would have moved backwards in time, resulting in conflicts on the constraint that this battery experiment should only be carried out in the sunlight.

overall preplanned experiment schedule. This could take place for example on cooling days. See Figure 6 for an example.

Due to the structure of the files exported from the Excel sheet, to the planning system it looked as if each of these experiments should be executed for 23.75 h every day. For other experiments that contained explicit On resp. Off procedures, these were scheduled at the beginning resp. the end of a given experiment opportunity. These opportunities were indicated by corresponding experiment resources allocated during the experiment execution. This would result in switching off the experiment at the end of the opportunity, which didn't correspond to the experimentator requirements, since the long-term experiments were supposed to run over several days. To ensure an experiment execution time for more than 24h, the constraints between experiment task and flight procedures were configured to merge these slots to an overall execution time of several days according to the original pre-planning schedule. This is possible, since all resource dependency types allow to specify offsets regarding start and end times.

In case of N17 this was sufficient since no further modification of the configuration of the experiment was necessary. The case of N19 was a bit more complicated since after a certain time of performing the experiment in one mode, the mode should change without switching the experiment off. To cope with this, the constraints were adapted in two ways:

1. The flight procedures to switch the experiment on had comparing resource dependencies that checked whether the corresponding experiment resource was 0 a configurable amount of seconds prior to the start of the experiment opportunity. This had to be checked irrespective of the mode that the experiment was previously running. If another experiment opportunity happens to lie before this one, the corresponding resource is set to 1 during this time. Therefore the flight procedure to switch on the experiment may not be allowed to be scheduled for the second experiment execution, since it is already running. Flight procedures to switch off the experiment were provided with comparing resource dependencies which checked that the resource is 0 after the end of the experiment execution. This prohibited the scheduling of the flight procedure when a follow up opportunity exists.

2. The flight procedures to configure the mode of a N19 long term experiment were only scheduled when the mode changed. Therefore it was sufficient, that these were provided with comparing resource dependencies to check the experiment was active in the same mode prior to the current experiment execution. If this was not the case the corresponding configuration flight procedure was scheduled.

For N17 first set of constraints was sufficient to schedule the tasks correctly, since only one mode was possible here.

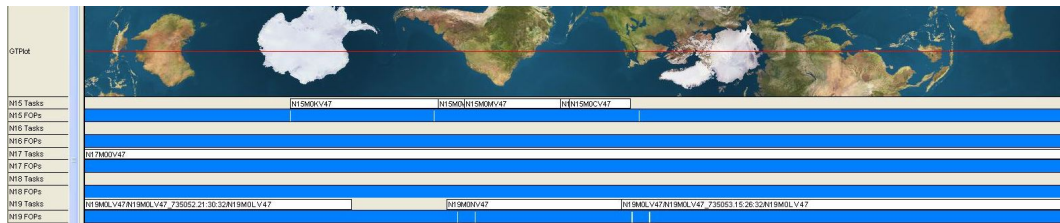


Figure 6. N15 and N19 operations. The top plot shows a ground track plot for the sub-satellite-point. The datatakes that previously, with the result of the pre-planning Excel file, would have taken place over uninteresting areas were moved to visibilities of specific regions of interest like South America. The plots below show experiment tasks and their scheduled flight procedures. Further down, the long term experiments N17 and N19 are plotted. N19 shows a significant gap. To continue the payload activity in this case, such gaps had to be merged by the MPS.

3. N15 datatake adjustments

At first, N15 IR camera experiments were fixed in time according to the overall pre-planning. This resulted in plenty of datatakes located over less interesting locations like oceans. To allow for more freedom in the choice of datatake acquisition regions, it was decided to shift the experiment execution within a certain time range.

A first flexibility gain was to allow moving the datatakes within the foreseen orbit of execution. It was therefore sometimes necessary to remove other experiments from the timeline, that were too close after the adjustment of the N15 experiment. A second one was enabled by placing the scenarios containing datatakes adjacent to cooling days. This allowed to shift the orbit as a whole. To guarantee the constraints from the pre-planning, it was necessary to shift the whole scenario, by the necessary number of orbits. The satellite was still nadir pointing at this time. Off nadir datatakes were not possible during the OOV phase of the mission. With these new degrees of freedom it was possible to give the camera investigators the possibility to gather data over regions of interest to them. Repositioning of the datatakes was subsequently done by hand and in close connection with the camera investigators. See Figure 6 on the previous page with an example in which a datatake was rescheduled to acquire data over South America.

VI. Conclusions and Outlook

The MPS for TET-1 during its OOV phase was presented. With its help it was possible to run the first year of operations in space. Due to its high flexibility, mainly based on the configurable model and algorithm composition, it was possible to react on problems that the mission experienced while the satellite was already orbiting in space.

For example, while camera datatakes before had only been scheduled once or twice a week, it was proposed to run a campaign with mainly N15 experiments in August 2013. To support more than the standard mode of the camera, the overall tele-command sequence was broken up into several parts, like switching on and off, gather GPS data, turn to earth pointing mode, perform a calibration datatake and configure the camera system according to the mode requested by the camera investigator. Furthermore, memory was now modeled for the first time, since each datatake allocates onboard memory differently according to its mode and duration. Additional downlink tasks freed the memory in the model according to the respective datatake's type. So the process of ordering datatakes had to be done with a lot of manual interaction.

To get rid of the process of placing datatakes on the timeline by hand, a complete order chain was developed for the follow-up mission phase, when TET-1 became part of FireBird.⁶ A user interface, the so-called Swath Preview and Ordering Tool (SPOT), now serves as a comfortable tool for the camera principle investigators to create and send datatake wishes, i.e. planning requests, after having had a preview on acquisition opportunities. These planning requests are imported into the newly designed MPS, which schedules the according datatakes and their downlinks, if feasible without conflicts.

Therefore a new set of scheduling algorithms was developed, that considers spacecraft resources like power thermal aspects by using gliding windows for the datatake execution and the memory of the datatakes. The splitting of downlinks for these datatakes over several downlink stations brings in an additional level of complexity for the scheduling algorithms.

In addition to fix standard datatake modes executed during the OOV mission phase, further camera modes are supported now. Additional interfaces to ground stations, containing information about scheduled datatakes, had to be set up. However the MPS for the FireBird mission relies still on the same set of basic algorithms that were already used during the OOV phase of TET-1. The background sequence algorithms only needed slight adaptations, while for the datatake acquisition new sequences of basic algorithms were composed, including new extensions to the set of basic algorithms. These were mostly necessary for the more difficult task to schedule the datatakes and their downlinks.

References

¹Axmann, R. and Wickler, M., "Development and Verification of an Autonomous On-Board Mission Planning System - an Example from the BIRD Satellite," *SpaceOps 2006*, Rome, Italy, June 06-23, 2006

²Axmann, R., Mühlbauer, P., Spörl, A., Turk, T., Föckersberger, S. and Schmolke, J. "Operations Concept for 11 different Payloads on the TET-1 Mini-satellite," *The 2010 4S Symposium - Small Satellite Systems and Services*, Funchal, Madeira, Portugal, May 31-June 4, 2010

³Mrowka, F., Geyer, M., Lenzen, C., Spörl, A., Göttfert, T., Maurer, E., Wickler, M., and Schättler, B., "The Joint

TerraSAR-X/TanDEM-X Mission Planning System”, Symposium Proceedings, pp. 3971–3974, *IGARSS 2011*, July 24-29, 2011, Vancouver, Canada, ISBN 978-1-4577-1004-9

⁴Wörle, M. T., Lenzen, C., Mrowka, F., Göttfert, T., Spörl, A., Grishechkin, B., and Wickler, M., “The Incremental Planning System - GSOC’s Next Generation Mission Planning Framework”, *SpaceOps 2014 13th International Conference on Space Operations*, 5–9 May 2014, Pasadena, California

⁵“Planning Modelling Language”,
URL:http://www.dlr.de/rb/en/Portaldata/38/Resources/dokumente/gsoc_dokumente/mb/GSOC_Modelling_Language.pdf

⁶Reile, H., Lorenz, E., and Terzibaschian, T., “The FireBird Mission – A Scientific Mission for Earth Observation and Hot SpotDetection,” *Small Satellites for Earth Observation, Digest of the 9th International Symposium of the International Academy of Astronautics*, Wissenschaft und Technik Verlag, ISBN 978-3-89685-574-9

⁷Lenzen, C., Wörle, M. T., Mrowka, F., Geyer, M. P., and Klaehn, R., “Automated Scheduling for TerraSAR-X/TanDEM-X,” *IWPSS 2011*, Darmstadt, Germany, June 08-10, 2011

⁸Lenzen, C., Mrowka, F., Spörl, A., and Klaehn, R. “Scheduling Formations and Constellations,” *SpaceOps 2010 11th International Conference on Space Operations*, 25–30 April 2010, Huntsville, Alabama